Weakly-Supervised Action Segmentation and Unseen Error Detection in Anomalous Instructional Videos

Reza Ghoddoosian Isht Dwivedi Nakul Agarwal Behzad Dariush
Honda Research Institute, USA
{reza_ghoddoosian, idwivedi, nakul_agarwal, bdariush}@honda-ri.com

Abstract

We present a novel method for weakly-supervised action segmentation and unseen error detection in anomalous instructional videos. In the absence of an appropriate dataset for this task, we introduce the Anomalous Toy Assembly (ATA) dataset[1], which comprises 1152 untrimmed videos of 32 participants assembling three different toys, recorded from four different viewpoints. The training set comprises 27 participants who assemble toys in an expected and consistent manner, while the test and validation sets comprise 5 participants who display sequential anomalies in their task. We introduce a weakly labeled segmentation algorithm that is a generalization of the constrained Viterbi algorithm and identifies potential anomalous moments based on the difference between future anticipation and current recognition results. The proposed method is not restricted by the training transcripts during testing, allowing for the inference of anomalous action sequences while maintaining real-time performance. Based on these segmentation results, we also introduce a baseline to detect pre-defined human errors, and benchmark results on the ATA dataset. Experiments were conducted on the ATA and CSV datasets, outperforming the state-of-the-art in segmenting anomalous videos under both online and offline conditions.

1. Introduction

One of the challenges in human machine interaction is the automatic vision-based understanding of human actions in instructional videos. These videos depict a series of low-level actions that collectively accomplish a top-level task, such as preparing a meal or assembling an object. However, labeling each frame of these videos can be arduous and necessitate a significant amount of manual effort to note the start and end times of each action segment. Consequently, there has been a surge of research interest in developing weakly-supervised methods to learn the actions. In particular, such methods aim to overcome the challenge of weakly-labeled instructional videos, where only the ordered sequence of action labels (**transcript**) is provided without any information on the duration of each action.

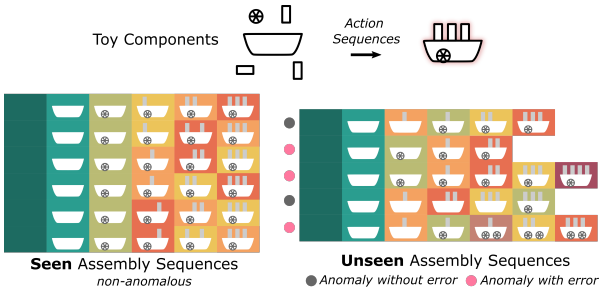

Figure 1. A toy example of different assembly sequences. Given a fixed set of non-anomalous training transcripts, our method explores and infers unseen anomalous sequences at test time. The anomalous sequences may or may not entail assembly errors.

Specifically, detection of fine errors and anomalies in tasks performed by human operators is critical for enhancing quality of work, safety, and efficiency. Anomalies can take different forms, but in this paper, we focus on the detection of fine-grained sequential anomalies in instructional videos. We define sequential anomalies as unseen action sequences that arise due to unexpected permutations of the action sequences seen in the training set. Such permutations may include unexpected changes in the order of actions, or the omission or addition of one or multiple actions at any point in the video. We also define an "*error*" as a sequential anomaly that leads to an undesired outcome. This means that not all sequential anomalies indicate faulty procedures (Fig.1). In the context of assembly, some examples of these unseen variations at test time include scenarios where an assembly worker skips fastening a screw or spends too much time idling between actions. AI systems can be trained on limited data collected from the optimal work of professionals, yet they must still be capable of detecting out-of-sequence actions in situations where inexperienced workers make mistakes or follow sub-optimal sequences.

Existing weakly-supervised segmentation methods di-

[1] https://usa.honda-ri.com/ata

vide a video into its constituent atomic actions. These methods are either explicitly [8, 15, 21] or implicitly [33] limited by the training transcripts, so they cannot identify anomalies and out-of-sequence actions effectively in videos. Moreover, the absence of datasets with anomalous instructional videos has prevented such methods from being thoroughly tested under circumstances, where there are significant variations between the training and test transcripts.

This paper presents the Anomalous Toy Assembly (ATA) dataset, which is the first instructional video dataset that contains anomalies. The ATA dataset comprises 1152 untrimmed videos of 32 participants assembling three different toys, which were recorded from four different viewpoints. Each toy assembly constitutes a task. The distinguishing feature of this dataset is the disparity between the training and test action sequences, enabling the investigation of unseen error detection in instructional videos.

In particular, the training set comprises 27 participants who assemble toys in an expected and consistent manner. On the other hand, the test and validation sets entail 5 participants who display anomalies in their assembly of toys, including sequence variations, defects, or redundancies. Although the test and training sets involve the same set of actions, the action sequences of the test set are distinct and previously unseen in the training set. Our dataset includes annotations of human errors, atomic actions, human pose, and bounding boxes of interactive objects for each video.

As a second contribution, we propose a novel method for weakly-supervised action segmentation, and unseen error detection in anomalous instructional videos. Unlike previous work, the proposed real-time action segmentation algorithm is not restricted by training transcripts during testing. Our inference method is a generalized version of the Viterbi algorithm [37] used in [14, 27]. It segments videos into unseen transcripts and identifies potential anomalous moments based on the difference between future anticipation and current recognition results. This method enables the inference of anomalous action sequences while maintaining real-time performance. Finally, based on these segmentation results, the paper introduces a mechanism to detect predefined human errors that occur during assembly, with baseline error detection results presented on the ATA dataset.

To summarize, the contributions of this paper are:

1) We introduce the ATA dataset, the first anomalous instructional video dataset, containing sequential anomalies and human errors made during toy assembly. All videos are annotated with various spatial and temporal labels.

2) We propose our unconstrained Viterbi algorithm that allows real-time segmentation of videos into unseen action sequences.

3) Experiments were conducted on the ATA and CSV [25] datasets, demonstrating that our proposed method outperforms the state-of-the-art (SoTA) in segmenting anomalous videos under both online and offline conditions. Here, online refers to the causal inference of the current action in streaming videos, while offline refers to the full segmentation of the video *after* observing the entire video.

2. Related Work

2.1. Weakly-Supervised Action Segmentation

The majority of weakly-supervised action segmentation methods [7, 8, 17, 21, 23, 27, 32, 33] focus on offline processing, and are limited by the training transcripts. Specifically, [7, 8, 21, 23, 26, 27, 32] cannot predict unseen transcripts, because they iterate through training transcripts to find the best alignment with the test video. In contrast, [33] trains an RNN to predict the video transcript offline, but the RNN remains biased by the training transcripts and unable to generalize well to unexpected transcript variations. In addition to the offline methods, [14] recently proposed a baseline for weakly-supervised online action segmentation, but its inference results are still constrained by the training transcripts. To this end, we purpose an unconstrained Viterbi algorithm. Our method is not only able to segment videos in a real-time and online manner, it also recognizes anomalous sequences and errors more effectively. We also compare our method with a greedy baseline [13] that is unconstrained and able to predict any combination of actions.

2.2. Anomaly and Error Detection

In the context of instructional videos, [31] classifies if a segment is a mistake or not, and [25] briefly addresses early warning. However, they study seen mistakes or assume availability of a known reference procedure respectively. Previous anomaly detection methods mainly focus on surveillance cameras. Based on the availability of anomalous events during training, these works are divided into open-set [1, 44], unsupervised [10, 36, 40, 43], weakly-supervised [19, 24, 42] and One Class Classification (OCC) [3, 20, 28, 39] methods. OCC methods, similar to ours, train on only normal videos, but they study anomaly differently. Firstly, they address binary detection of anomalies versus the non-anomalous background in surveillance videos. Meanwhile, we focus on segmenting instructional videos into unexpected sequences of fine-grained actions and identifying the resulting unseen error classes. Secondly, they define anomaly by occurrence of a visually out-of-distribution event, while anomaly in our work is defined as "unseen" permutations of "seen" actions.

2.3. Instructional Video Datasets

Some instructional video datasets [2, 11, 35, 45] are collected from YouTube, containing edited footage with large background segments. Meanwhile, most efforts for real-time recording of tasks and their atomic actions are in domains of cooking [16, 29, 34] and assembly [5]. Some

Table 1. Comparison of real-time instructional video datasets. The overline denotes average. * [31] indicates only if a segment is a mistake or not. Results of [31] based on long disassembly-assembly videos.

Perspective	Dataset	#Vids	Dur.	#Tasks	#Actions	#Transcripts	Transc. len.	Vid dur.	#Views	Error labels	Frame labels	Pose	Obj. bb
1^{st} Person	GTEA [12]	28	0.6h	7	11	28	33	1.2m	1	×	✓	✓	×
1^{st} Person	CSV [25]	1940	11.1h	14	18	70	9.5	0.3m	1	×	×	×	×
1^{st} Person	1ReC [30]	450	27h	15	102	418	11.7	3.6m	1	×	✓	×	×
1^{st} Person	ASM101 ego [31]	1425	167h	101	202	362	24	7.1m	4	×*	✓	✓	×
3^{rd} Person	Cooking2 [29]	273	8h	58	87	272	95.5	6m	1	×	✓	✓	×
	50 Salad [34]	50	4.5h	1	19	50	20	6.4m	1	×	✓	×	×
	Breakfast [16]	1712	77h	10	47	256	6.9	2.3m	2-5	×	✓	×	×
	IKEA [5]	1113	35.3h	4	33	359	22.7	1.9m	3	×	✓	✓	✓
	3ReC [30]	1349	82h	15	102	441	11.7	3.6m	3	×	✓	×	×
	ASM101 [31]	2896	346h	101	202	362	24	7.1m	8	×*	✓	×	×
3^{rd} Person	**ATA**	1152	24.8h	3	15	141	12.9	1.3m	4	✓	✓	✓	✓

datasets [9, 22] are studied as shorter trimmed clips due to their very long original videos. Among untrimmed datasets, [29, 34] are small and lack sufficient training samples. Other datasets such as [5, 12, 16, 30] have similar and error-free action sequences within the same task unlike our proposed dataset. Therefore, these datasets are not suitable for studying error detection and action segmentation of anomalous sequences. In ASM101 [31] action segments are labeled as one of correct, mistake and correction categories, where such segments exist in both training and test sets. In contrast, in our ATA we have unseen test errors categorized into 11 specific classes, so there is a larger discrepancy between test and training sequences. Recently, the CSV dataset [25] was created to study 14 different tasks in chemical experiments with 5 unique transcripts per task. [25] is an egocentric dataset, and introduces diverse transcripts across different tasks. However, it lacks temporal annotations and errors, and has limited intra-task variations. Table1 shows a comparison of existing datasets.

3. Anomalous Toy Assembly (ATA) Dataset

This section introduces the ATA dataset, the first public dataset to study sequential anomalies in instructional videos. For additional details, see the appendix.

3.1. Data Collection

32 volunteers assembled three toys (*airplane*, *table*, and *record player*) in a lab environment. Each participant completed each assembly three times, resulting in nine total sequences per person for the three tasks. Four ZED Mini cameras recorded the participants from four viewpoints (front, side, overhead, and global). Participants' faces were blurred, and audio was muted for privacy reasons. See Fig. 2 for the camera viewpoints.

Action transcripts were scripted before recording, and participants were briefed on the transcript, object names and action definitions. During recording, participants followed our verbal instructions for the next action, enabling seamless completion of each scripted assembly task.

3.2. Data Statistics

Overview: The ATA dataset includes 1152 untrimmed RGB videos, totaling 24.8 hours, with a resolution of 1920×1080 and a frame rate of 30 fps. Videos were recorded by four cameras during nine sessions per participant, resulting in a mean duration of 1.3 mins and a range of 0.6 to 2.1 mins. The dataset contains 15 atomic actions such as "*fasten screw*" and "*take plate*" and 11 error classes. Consequently, there are 141 unique action sequences with an average length of 12.9 action segments per video. The gender ratio of the 32 participants is 3 to 1 male to female. See Fig.2 for the occurrence distribution of action and error labels.

Data Splits: We split the participants into training, validation, and test sets consisting of 27, 1, and 4 participants, respectively. The validation and test sets include videos with sequential anomalies, defined as unexpected permutations of the training transcripts, such as redundant actions (*e.g.*, inserting an extra screw or unexpected background segments between actions), skipped actions (*e.g.*, not tightening a screw), and major changes in the order of training action subsequences (*e.g.*, when the last phase of an assembly is done in the beginning of the video).

The dataset includes 96 training transcripts and two separate sets of 9 validation and 36 testing transcripts that are mutually exclusive from the training set. While all sets differ in transcripts, they share the same tasks and actions. This is in contrast to standard splitting of [25], where test and training sets come from entirely different tasks, so its transcripts set entails cross-task variations rather than intra-task anomalies.

3.3. Data Annotations

Temporal Action Labels: We have annotated the start and end frames of all action segments in each video in addition to the action transcripts. To ensure consistency among annotators regarding action definitions, three experts temporally annotated all 15 atomic actions.

Video-Level Error Labels: Although all test videos involve sequential anomalies, some of these videos still

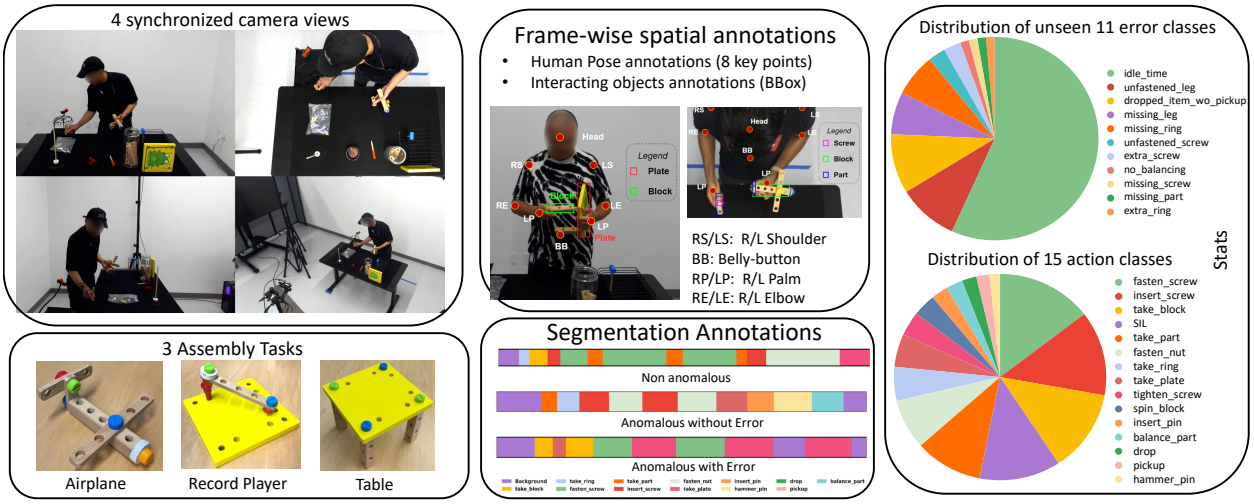

Figure 2. Overview of the ATA dataset

demonstrate a valid and complete assembly sequence. Consequently, we identified and labeled the unseen errors that occurred during the assembly of each toy in the test set. There are a total of 11 error categories, including *unfastened screw*, *idle time*, and *not picking up a dropped item*. Each test video is annotated with one or more video-level error labels that indicate which error classes are present in the video. Our intention is to encourage further research on error detection in instructional videos.

Object Bounding Boxes: Annotation workers were instructed to label and provide bounding boxes of all objects that each participant touches for all frames.

Human Pose: The workers also provided labels for 8 upper body joints for all frames.

These annotations, along with the object bounding boxes, provide human-object interaction information that can be used for future research on the ATA dataset.

4. Action Segmentation of Anomalies

In this section, we present our segmentation model and propose the unconstrained Viterbi algorithm for detecting anomalies and unseen action sequences during testing, as our method focuses more on the test phase due to the absence of anomalous sequences during training. The training strategy is discussed at the end of the section.

4.1. Problem Definition

Given a set of training videos $\mathbb{V}$ and corresponding transcripts $\mathcal{T}_s$, the goal is to partition a test video into a sequences of n actions $\boldsymbol{a}_1^n \in \mathcal{T}_u$ and their duration $\boldsymbol{l}_1^n$. $\mathcal{T}_u$ is the set of unseen test transcripts, and based on the anomaly assumption, $\mathcal{T}_u \cap \mathcal{T}_s = \emptyset$. We define $\mathbb{A}$ as the set of $|\mathbb{A}|$ unique actions labels and $\mathbf{x}_1^t$ the sequence of frame-level features from the beginning of video until time t.

4.2. Proposed Inference Method

Inference Model Overview: Given trained parameters and extracted features $\mathbf{x}_1^t$ of a test video, we use Eq.1 to approximate the likelihood of n action segments with labels $\boldsymbol{a}_1^n$ and durations $\boldsymbol{l}_1^n$ until time t. In Eq.1, $p(x_t|a_{n_t})$ is the visual model, and is derived using the Bayes rule on top of the probability output of a recognition network as in [21]. n_t is the segment number at time t.

We also model the probability of transitioning into action segment $\acute{n} > 1$ at time $t_{\acute{n}} = \sum_1^{\acute{n}-1} l_{\hat{n}}$ by $p(a_{\acute{n}}|\boldsymbol{a}_1^{\acute{n}-1}, \mathbf{x}_1^{t_{\acute{n}}})$. Our proposed action transition model depends on the context of video at the transition point $t_{\acute{n}}$ in addition to previous action labels $\boldsymbol{a}_1^{\acute{n}-1}$. This helps to detect anomalous segments unlike previous work, where action transition occurs only according to the set of seen training transcripts $\mathcal{T}_s$. Finally, $p_{\text{mode}}^{\acute{n}}(l|a)$ is the length model of segment $\acute{n}$, and estimates the probability of action a lasting l frames.

$$p_{\text{mode}}(\boldsymbol{a}_1^n, \boldsymbol{l}_1^n|\mathbf{x}_1^t) \approx \prod_{\acute{t}=1}^{t} p(x_{\acute{t}}|a_{n_{\acute{t}}}) \prod_{\acute{n}=1}^{n} p_{\text{mode}}^{\acute{n}}(l_{\acute{n}}|a_{\acute{n}}) p(a_{\acute{n}}|\boldsymbol{a}_1^{\acute{n}-1}, \mathbf{x}_1^{t_{\acute{n}}}). \quad (1)$$

Eq.1 addresses segmentation for both **modes** of offline (off) and online (on). In offline segmentation t marks the end of a_n. Meanwhile in the online mode, the last segment n is ongoing, so t may not mark the end of the current action. Hence, the difference between both modes is the choice of the length model $p_{\text{mode}}^n(l|a)$ for the last segment n. While $p_{\text{off}}^{\acute{n}}(l|a)$ is a Poisson function in offline segmentation for all segments, in the online mode we use half Poisson to model $p_{\text{on}}^n(l|a)$ for only the last segment n [14]. All Poissons are parameterized by the estimated average length of actions.

Action Transition Model: Let $\vec{\boldsymbol{A}}_t = [p(c_t|x_{t-\omega}; \theta_a), \forall c \in \mathbb{A}]$ be the "anticipated" action probability vector of time t given past features at time

$t-\omega$. Also, $\vec{\boldsymbol{R}}_t = [p(c_t|x_t;\theta_r), \forall c \in \mathbb{A}]$ denotes the "current" action probability vector. $\vec{\boldsymbol{A}}_t$ and $\vec{\boldsymbol{R}}_t \in \mathbb{R}^{|\mathbb{A}|}$ are outputs of anticipation and recognition networks, parameterized by θ_a and θ_r respectively.

We detect anomalous behavior by the discrepancy between the expected and current action representations $\vec{\boldsymbol{A}}_t$ and $\vec{\boldsymbol{R}}_t$ respectively. Additionally, it has been shown [18] that actions that typically occur temporally close to each other, are also more similar in their visual representations. We exploit these properties in Eq.2 to connect similarity between action representations to their temporal positions. Specifically, let $\mathbb{E}_\tau(\boldsymbol{a}_1^{\acute{n}-1}, t_{\acute{n}})$ be the set of possible action labels for segment $\acute{n}$ at transition point $t_{\acute{n}}$ given the previous labels $\boldsymbol{a}_1^{\acute{n}-1}$. In Eq.2, $\mathfrak{s}()$ is the cosine similarity of two vectors, and $\mathcal{N}^+_{\mathcal{T}_s}\{\boldsymbol{a}_i^j\}$ returns the set of all actions that succeed sequence $\boldsymbol{a}_i^j$ according to the training transcripts $\mathcal{T}_s$.

$$\mathbb{E}_\tau(\boldsymbol{a}_1^{\acute{n}-1}, t_{\acute{n}}) = \begin{cases} \mathcal{N}^+_{\mathcal{T}_s}\{\text{LCS}(\boldsymbol{a}_1^{\acute{n}-1}, \mathcal{T}_s)\} & \text{if } \tau < \mathfrak{s}(\vec{\boldsymbol{R}}_{t_{\acute{n}}}, \vec{\boldsymbol{A}}_{t_{\acute{n}}}) \\ \mathbb{A} & \text{otherwise} \end{cases} \tag{2}$$

Dissimilar anticipated and current action probability vectors can indicate an anomalous transition, so we allow deviation from the transcripts by exploring the set of all possible actions $\mathbb{A}$. Otherwise, action transitions only follow the sequences in the training transcripts. In this case the set of all possible actions is equal to the set of all actions that succeed the Longest Common Subsequence (LCS) $\boldsymbol{a}_i^{\acute{n}-1}$ between the previous sequence $\boldsymbol{a}_1^{\acute{n}-1}$ and the training transcripts $\mathcal{T}_s$. Note that the transition model in previous work [21, 14] is a special case of ours when $\tau = 0$, because in this case deviation from training transcripts never occurs and the LCS is always $\boldsymbol{a}_1^{\acute{n}-1}$. Ultimately, in Eq.1, $p(a_{\acute{n}}|\boldsymbol{a}_1^{\acute{n}-1}, \mathbf{x}_1^{t_{\acute{n}}}) = 1$ if $a_{\acute{n}} \in \mathbb{E}_\tau(\boldsymbol{a}_1^{\acute{n}-1}, t_{\acute{n}})$, and it is 0 otherwise.

Unconstrained Viterbi Algorithm: We propose Algorithm1 to efficiently solve both online and offline segmentation of Eq.1 at each time step t. At each time step t, we use dynamic programming and the results of previous time step to generate new segmentation results. Each new sequence is the result of either continuing the last action or transitioning into a new one. Different than the Viterbi algorithm of [27], our Viterbi algorithm is unconstrained, because it is not limited to the training transcripts. This is essential for inferring unseen and anomalous action sequences, as our method can theoretically generate any sequence of actions. Specifically, $P_t[l_n, \boldsymbol{a}_1^n]$ is defined as the probability of the most likely alignment of sequence $\boldsymbol{a}_1^n$ with video frames until time t, so that a_n is incomplete and has a duration of l_n. The mostly likely segmentation result $(\overline{\boldsymbol{a}}_1^n, \overline{\boldsymbol{l}}_1^n)_{\text{mode}} = \text{argmax}\{P_t[\acute{l}_{\acute{n}}, \acute{\boldsymbol{a}}_1^{\acute{n}}] \cdot p^{\acute{n}}_{\text{mode}}(\acute{l}_{\acute{n}}|\acute{a}_{\acute{n}})\}$ is derived once at the end of video for offline segmentation, and at every time step during online inference. However, we pick only the current action $a_t = \overline{a}_n$ as the online inference output of time t.

In order to achieve a real time performance, separately for each action $a \in \mathbb{A}$ at time t, we keep only the set of top B likely segmentation results $\mathcal{S}_t^B(a)$ ending with action a. Such an action-wise pruning gives the online segmentation method the advantage to infer any possible action, which might have been pruned out otherwise. The overall complexity of Alg.1 at each time step is $O(B|\mathbb{A}|(\log B + |\mathbb{A}|))$. This complexity is the result of enumerations in addition to the sorting complexity of $top^B\{\}$ with beam size B.

Algorithm 1 Unconstrained Viterbi Algorithm at time t

Input: Video features $\mathbf{x}_1^t$, and past results: P_{t-1} and $\mathcal{S}_{t-1}^B$
Output: P_t, $\mathcal{S}_t^B$ and the current action $\overline{a}_n$

for $a \in \mathbb{A}$ **do**:
  for $(\boldsymbol{a}_1^n, \boldsymbol{l}_1^n) \in \mathcal{S}_{t-1}^B(a)$ **do**:
    $P_t[l_n + 1, \boldsymbol{a}_1^n] = P_{t-1}[l_n, \boldsymbol{a}_1^n] \cdot p(x_t|a)$
    for $a_{n+1} \in \mathbb{E}_\tau(\boldsymbol{a}_1^n, t)$ **do**:
      $Q[l_n, \boldsymbol{a}_1^{n+1}] = P_{t-1}[l_n, \boldsymbol{a}_1^n] \cdot p(l_n|a) \cdot p(x_t|a_{n+1})$
  $\forall \boldsymbol{a}_1^n : P_t[1, \boldsymbol{a}_1^n] = \max_{\acute{l}_{n-1}}\{Q[\acute{l}_{n-1}, \boldsymbol{a}_1^n]\}$
$\overline{a}_n = \underset{\acute{a}_{\acute{n}} \in \mathbb{A}}{\text{argmax}}\{\max_{\acute{l}_{\acute{n}}, \acute{\boldsymbol{a}}_1^{\acute{n}-1}}\{P_t[\acute{l}_{\acute{n}}, \acute{\boldsymbol{a}}_1^{\acute{n}}] \cdot p^{\acute{n}}_{\text{on}}(\acute{l}_{\acute{n}}|\acute{a}_{\acute{n}})\}\}$
$\forall a \in \mathbb{A} : \mathcal{S}_t^B(a) = \left\{(\acute{\boldsymbol{a}}_1^{\acute{n}}, \acute{\boldsymbol{l}}_1^{\acute{n}}) \mid P_t[\acute{l}_{\acute{n}}, \acute{\boldsymbol{a}}_1^{\acute{n}}] \in top^B\{P_t[:,:]\} \wedge \acute{a}_{\acute{n}} = a\right\}$
return $\mathcal{S}_t^B, P_t, \overline{a}_n$

4.3. Weakly-Supervised Training

We use the weakly-supervised framework in [21] to train our anticipation and recognition networks iteratively. Given a video of length T and its transcript per iteration, training is done following two steps. First, frame-level pseudo labels $\overline{\boldsymbol{a}}_1^T$ are estimated through offline segmentation in Eq.1. Second, the pseudo labels are used in a loss function $\mathcal{L}$ to update the parameters θ_a and θ_r of the anticipation, and recognition networks respectively. Particularly, we employ the Constrained Discriminative Forward Loss $\mathcal{L}_{\text{CDFL}}$ [21], which effectively maximizes the decision margin between the valid and hard invalid pseudo labels. In Eq.3, we apply $\mathcal{L}_{\text{CDFL}}$ to the recognition outputs $\vec{\boldsymbol{R}}_1^T$ of all frames and to the anticipation output $\vec{\boldsymbol{A}}_\omega^T$, weighted by λ_a, for frames from ω to T, where ω is the future anticipation range.

$$\mathcal{L} = \mathcal{L}_{\text{CDFL}}(\overline{\boldsymbol{a}}_1^T, \vec{\boldsymbol{R}}_1^T) + \lambda_a \mathcal{L}_{\text{CDFL}}(\overline{\boldsymbol{a}}_\omega^T, \vec{\boldsymbol{A}}_\omega^T). \tag{3}$$

5. Error Detection in Instructional Videos

Problem Definition: The goal in error detection is to identify the number of times n_e an error $e \in \mathcal{E}$ has occurred in a test video. $\mathcal{E}$ is the set of unseen error categories that are only present in the test set. The dataset provides detailed instructions I of what constitutes each error when performing a task, *e.g.*, the error label "*Missed Leg*" means using less than 4 legs to assemble a table. It is not clear how to temporally locate all errors because certain errors correspond to inaction. Also, some errors can only be inferred when the video has ended, *e.g.* not picking up an item that is dropped

in the process. As a result, we detect errors at the end of the video after the task is fully observed.

Method Overview: We propose a simple error detection method as a set of error functions $\{\mathcal{F}_e\}$, so that each function $\mathcal{F}_e : \mathbf{f} \xrightarrow{I} n_e$ maps frequency $\mathbf{f}$ of inferred actions in the test video to the number of instances n_e that error e has occurred. Here $\mathbf{f} = \{\mathrm{f}_a\}$, and f_a is the number of predicted video segments labeled by action a. For example, the function for the error label "*Unfastened screw*" is defined as $\mathcal{F}_{\textbf{Unfastened screw}} := max(\mathrm{f}_{\text{insert screw}} - \mathrm{f}_{\text{fasten nut}}, 0)$. Below we go over the details of our baseline:

Method Details: For each test video, we first generate two different segmentation results S^0 and S^τ for $\tau = 0$ and $\tau > 0$ respectively. The former represents the constrained offline segmentation as a reference, where the estimated transcript is one of the training transcripts. Then, the respective set of action frequencies $\mathbf{f}^0$ and $\mathbf{f}^\tau$ are calculated from the segmentation results S^0 and S^τ. Finally we incorporate $\mathbf{f}^0$ and $\mathbf{f}^\tau$ in Eq.4 to produce if and how many times each error e has happened:

$$n_e = \mathcal{F}_e(\mathbf{f}^\tau) \times \underbrace{\big(1 - min(\mathcal{F}_e(\mathbf{f}^0), 1)\big)}_{b}. \tag{4}$$

Error functions operate based on action frequencies and do not consider the semantics of the video. Therefore, we use the reference action frequency $\mathbf{f}^0$ to focus on only relevant errors and alleviate false positives. Specifically, term b in Eq.4 conditions the result based on the action frequency discrepancy between the predicted anomalous transcript and its corresponding non-anomalous training transcript. In other words, we detect an erroneous behavior in the anomalous segmentation result only if the same behavior is error-free in the estimated non-anomalous transcript of the video, *e.g.*, skipping action a in a test video is an error only if action a has occurred in its non-anomalous reference S^0.

6. Experiments

Datasets. The *CSV* dataset [25] consists of 14 diverse chemical tasks, where each task has 5 unique transcripts. Following their task-wise splitting, we use the first two tasks for testing, the third task for validation and the remaining 11 tasks for training. Such splitting imposes variations between test and training transcripts for our study. Refer to table 1 for more statistics. We also show results on the *ATA* dataset following our standard splitting.

Metrics. Based on previous segmentation works [21, 33, 4], we use 4 complementary metrics to evaluate our action segmentation results: 1) *acc* as the mean frame-wise accuracy. 2) *IoU* computes the intersection over union for each predicted segment. 3) *F1@0.5* measures the class-wise F1 score of predicted segments using a 0.5 *IoU* threshold. 4) *edit score* utilizes edit distance to evaluate the similarity between the predicted and ground-truth transcripts. Results on the CSV dataset are reported using only the *edit score* due to their lack of frame-wise annotations. Additionally, we propose $F1_{\text{error}}$ to evaluate false and true positive rates in error detection through class-wise F1 score.

Table 2. Comparison of our method with various online and offline action segmentation baselines on ATA and CSV datasets.

		ATA Dataset(%)					CSV (%)
Mode	Method	acc	IoU	edit	F1@0.5	$F1_{\text{error}}$	edit
Online	Greedy [13]	62.3	54.0	51.1	44.6	53.0	56.4
	DP_{con} [14]	56.4	45.1	58.5	44.7	41.0	57.8
	Ours	**62.8**	**54.3**	**58.5**	**49.6**	**56.0**	**58.1**
Offline	MuCon [33]	52.6	42.7	37.3	24.0	43.5	58.6
	TASL [23]	40.5	27.7	57.3	27.1	0	53.0
	CDFL [21]	59.2	45.5	60.0	51.9	0	54.2
	Ours	**66.1**	**57.7**	**68.8**	**62.0**	**59.2**	**60.3**

Implementation Details. We employ the pre-trained I3D network [6] to extract features from optical flow [41] and RGB frames for both datasets. As our neural network, we train the Transformer encoder of [38] with two separate recognition and anticipation tokens, and use λ_a of 0.1. Further, we set B to 150, and ω to 1 sec. The threshold τ is chosen based on the validation set. Specifically, τ is 0.7 for all experiments except for offline segmentation of CSV videos, where $\tau = 0.3$. We evaluate the sensitivity of results to τ in Sec.6.2. All experiments are based on a fixed random seed and following previous work [21, 33, 23], results correspond to inferring action sequences every 30 and 15 frames for the ATA and CSV datasets, respectively. Finally, our defined set of error functions, error instructions I, and all parameters are provided in the appendix.

6.1. Comparison with State of the Art

6.1.1 Online Segmentation

Baselines. 1) *Greedy* [13]: We train this baseline with our loss function and Transformer for a direct comparison. At test time, it directly uses the Transformer output to predict the current action following a sliding window approach. 2) We compare with the single-view DP_{con} [14], which utilizes constrained dynamic programming (DP) to predict the current action based on training transcripts. We used our Transformer for DP_{con} as well.

Comparison. The Greedy baseline suffers from over-segmentation, which is indicated by low *edit* and *F1* values. Meanwhile, our method can be viewed as a hybrid form of the Greedy and DP_{con} approaches. Particularly, it follows the action precedence in the training transcripts but also, like the Greedy approach, explores all action possibilities when a potential anomaly is indicated, *i.e.*, $\tau > \mathfrak{s}(\vec{\boldsymbol{R}}_t, \vec{\boldsymbol{A}}_t)$. Table 2 reflects this property, where our proposed method is able to produce the superior online results of the two baselines in *acc*, *IoU* and *edit score*, while outperforming both significantly in $F1_{\text{error}}$ and *F1@0.5*. For the CSV dataset,

our edit score is only slightly better than [14]. However, comparison of online results for the ATA dataset suggests a potentially larger superiority of our method in other metrics for the CSV dataset too.

6.1.2 Offline Segmentation

Baselines. We evaluated our offline segmentation results by comparing them with other open source SoTA methods. Among them, CDFL [21] is the most similar to our approach, and it is the offline inference version of the constrained DP in [14]. We therefore trained CDFL using our settings and Transformer. Additionally, we used TASL [23] and MuCon [33] as other comparison points. While the segmentation result of TASL [23] is limited by the training transcripts, MuCon [33] has the ability to predict previously unseen transcripts during testing.

Comparison. Compared to online segmentation, our offline results outperform the previous work more significantly in all metrics and datasets. We explore the reason more in Sec. 6.1.4. Although [33] is able to predict unseen transcripts, its *edit score* is inferior to ours on both datasets. This highlights the generalization limitations of [33] to unexpected transcript changes in case of sequential anomalies.

6.1.3 Error Detection

Unseen error detection can be considered as a semantic way of evaluating and explaining the segmentation results in anomalous instructional videos. As such, we have applied our proposed error detection method to the segmentation results of all baseline methods to compare their ability to detect assembly errors in the ATA dataset (Table2). For a fair comparison, we have used our reference segmentation S^0 in all methods. Unsurprisingly, constrained offline methods, *e.g.*, [21, 23], have an $F1_{\text{error}}$ value of 0, because their predicted transcript is from the error-free training set, resulting in 0 true positives. It is worth noting that while DP_{con} uses the training transcripts to infer the most likely action sequence at each time step, its resulting segmentation can still correspond to an unseen transcript since predictions at different time steps are independent of each other. Despite this advantage, DP_{con} remains unable to successfully predict anomalous sequences and errors.

6.1.4 Semi-Online Segmentation

The amount of delay in segmenting videos can range from no delay (online) to the entire duration of the video (offline). When video is segmented with a delay within this range, it is referred to as semi-online segmentation [14]. Fig. 3 compares our semi-online segmentation with previous work using 4 metrics on the anomalous videos of the ATA dataset.

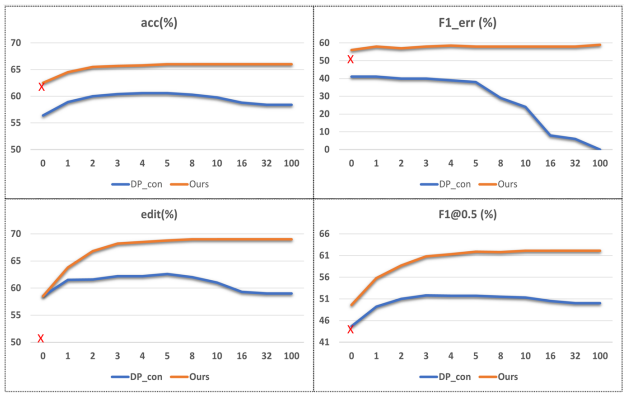

Figure 3. Semi-online segmentation results of DP_{con} [14] and ours given the delay in seconds. The red cross indicates the online Greedy [13] result.

The accuracy of our unconstrained segmentation monotonically improves with longer delays because the added delay provides more information and results in correcting earlier inferred action segments. On the contrary, with more delay the performance of [14] initially increases but it quickly drops because it starts to lose its ability in predicting unseen sequences as it transitions from constrained online to constrained offline segmentation. This is best observed in $F1_{\text{error}}$, where the rate of detected errors approaches 0 as the latency converges to the offline limit.

6.2. Sensitivity Analysis

Threshold Sensitivity. The threshold τ determines how often our Viterbi algorithm explores all action possibilities beyond the transcripts constraint. τ ranges from 0 to 1 and it changes the hybrid tendency of our algorithm from no exploration of fully constrained to constant exploration of Greedy. In online segmentation of anomalous videos on the ATA dataset Greedy predictions (dashed lines in Fig.4) are more reliable than the constrained baseline ($\tau = 0$), so frequent exploration ($\tau > 0.6$) leads to fairly stable and better results. This interval is shaded in Fig.4 (left). However, increasing τ does not always lead to a monotonically better performance, as both *F1* and *acc* have a slight drop for $\tau > 0.8$ due to false positives. On the other hand, on the

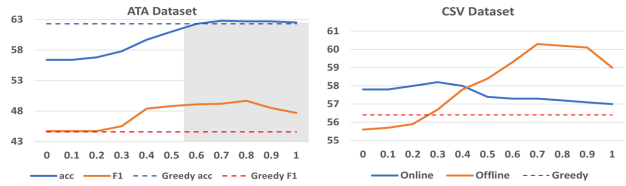

Figure 4. Left: Effect of the threshold τ on online segmentation results of the ATA dataset based on *acc* and *F1@0.5* metrics. Right: Comparison of online and offline segmentation *edit scores* under different thresholds in the CSV dataset.

Table 3. The effect of beam size and exploration threshold τ in the performance on the ATA dataset. Inference done every 30 frames.

	$B(\tau = 0.7)$				$B(\tau = 0.2)$			
Metric	1	10	150	1k	1	10	150	1k
Online acc %	62.4	62.7	62.8	62.8	49.0	54.0	56.8	56.8
Offline acc %	65.7	65.9	66.1	66.1	50.5	55.7	59.8	60.0
Average Online fps	54k	6.1k	310	38	105k	15k	900	122

Table 4. *edit score* of constrained and unconstrained offline segmentation with varying degrees of anomaly on the CSV dataset.

	Seen transcript set $\mathcal{T}_s$			
Mode	$\mathcal{T}_{\text{all}}$	$\mathcal{T}_{\text{tr}} \cup \mathcal{T}_{\text{v}}$	$\mathcal{T}_{\text{tr}}$	$\mathcal{T}_{\text{tr}}^{\subset}$
Constrained Viterbi ($\tau = 0.0$)	65.8	55.8	55.7	53.8
Unconstrained Viterbi ($\tau = 0.7$)	62.0	60.3	60.3	59.6

CSV dataset in Fig.4 (right), Greedy predictions are inferior to the constrained baseline, so our online method tends to perform best when exploration is limited ($\tau < 0.4$). We also compare the online and offline segmentation results on the CSV dataset. Interestingly, our method in the offline mode compensates for the unreliability of greedy predictions and performs best with high thresholds.

Beam Size Sensitivity and Real-Time Performance. Table 3 shows the effect of beam size B on our online and offline segmentations given a high and a low exploration threshold τ. Our online and offline segmentation results for $\tau = 0.7$ are hardly sensitive to B, which is mainly due to a high exploration rate τ in segmentation. This allows transitions between many confident action segments over time, so that the inferred sequences at each step are hardly punned out. On the contrary, segmentation with a low τ is constrained, so it cannot explore enough possibilities to increase its chance of staying in the beam. Consequently, it is less invariant to B. Additionally, Table.3 demonstrates that even with a high degree of exploration $\tau = 0.7$, our action-wise pruning can ensure a real-time fps. We measured fps based on the average time it takes to process a video.

Sensitivity to the Discrepancy between $\mathcal{T}_s$ and $\mathcal{T}_u$. Table 4 compares constrained and unconstrained offline segmentation under varying degrees of anomaly on the CSV dataset. Specifically, we use the discrepancy between seen and test transcripts as a metric to measure the anomaly degree. To this end, we create a subset of the original training transcript set $\mathcal{T}_{\text{tr}}$ by removing 2 of the 11 training tasks. $\mathcal{T}_{\text{tr}}^{\subset}$ denotes this subset of the training transcript set. We initialize the set of seen transcripts $\mathcal{T}_s$ at test time by the set of all transcripts $\mathcal{T}_{\text{all}}$ in the dataset, and gradually increase the degree of anomaly by limiting $\mathcal{T}_s$ to $\mathcal{T}_{\text{tr}} \cup \mathcal{T}_{\text{v}}$, $\mathcal{T}_{\text{tr}}$, and $\mathcal{T}_{\text{tr}}^{\subset}$, where $\mathcal{T}_{\text{v}}$ is the set of validation transcripts. Initially, the *edit score* for constrained segmentation is higher than that of unconstrained segmentation when using $\mathcal{T}_{\text{all}}$, since all transcripts are available and test sequences entail no unexpected anomaly to explore. However, as the discrepancy between seen and test transcripts increases, the unconstrained Viterbi outperforms the constrained Viterbi by a larger margin.

6.3. Additional Experiments and Ablation Study

We compare the performance of our proposed algorithm for seen vs. unseen transcript and provide additional experiments on how our unconstrained Viterbi is effective for fully supervised inference. Finally we conclude this section by doing an analysis on the error detection task and further results on the 50Salad[34] dataset.

6.3.1 Seen vs. Unseen Transcripts

At test time, each person assembles each toy 3 times. To present separate results on seen/unseen transcripts, we do 3 experiments, where the transcripts for one of the 3 recordings is seen at a time. Table 5 shows the average online segmentation results of these 3 experiments for test videos with seen and unseen transcripts. Results confirm the performance of the unconstrained Viterbi is more invariant to seen/unseen transcripts and generalizes better to unseen cases. However, unsurprisingly, constraining the results to seen transcripts leads to a better performance for videos with seen transcripts as there is no need to explore new variations.

6.3.2 Fully-Supervised Comparison

ATA dataset can be used for any form of supervision. Table 6 benchmarks one of the SOTA fully supervised online action detection methods (Oadtr [38]). Table 2 and Table 6 show that our unconstrained Viterbi can improve results in both weakly and fully supervised settings. Evidently, [38] performs best when combined with our unconstrained Viterbi and mitigates the over segmentation of greedy Oadtr predictions.

6.3.3 Error Detection Analysis

Out of the 144 test videos, 40 videos are error-free and 104 are erroneous. Each test video has a max of 5 error instances

Table 5. Results on ATA test videos w/ seen and unseen transcripts.

	Seen / Unseen / Mixed transcripts			
Mode	acc(%)	IoU(%)	edit(%)	F1_{err}(%)
Constrained Vite. $\tau = 0.0$	71/57/62	60/46/51	76/58/65	72/49/56
Unconstrained Vite. $\tau = 0.7$	65/64/64	55/55/55	58/57/57	58/57/57

Table 6. Fully supervised online segmentation on the ATA dataset.

Mode	Method	acc(%)	IoU(%)	edit(%)	F1@0.5(%)
Full	Oadtr + Cons. Viterbi	59.1	49.5	56.6	47.9
Full	Oadtr + Uncons. Viterbi	65.5	58.9	54.1	51.0
Full	Greedy Oadtr[38]	66.0	58.7	49.0	46.1

Table 7. Online segmentation of erroneous and error-free videos.

Mode	acc(%)	IoU(%)	edit(%)	F1@0.5(%)
Error-free videos	55.8	48.9	52.4	44.9
Erroneous videos	66.0	56.1	58.5	50.7

Table 8. Weakly supervised online segmentation (50Salad-split1).

Method	acc(%)	IoU(%)	edit(%)	F1@0.5(%)
Greedy	39.3	25.1	14.2	5.3
DP_{con}	41.7	29.3	33.9	18.2
Ours ($\tau = 0.2$)	44.0	31.6	34.5	18.2

with an average of 1.7 unique errors. Table 7 shows online segmentation results of erroneous and error-free videos. Interestingly, the results of the erroneous videos are better. Note that "error-free videos still include unseen sequential anomalies and variations", e.g., assembling pieces in a reverse order to those of training samples. In our test set, it *happens* that error-free videos present more challenging sequential and visual variations. so results depend on the degree of sequential and visual variations and not necessarily presence of error.

6.3.4 Online Segmentation on the 50 Salad Dataset

Our work focuses on the study of anomalous instructional videos, which limits suitable existing datasets that we can use. Breakfast, GTEA and IKEA datasets follow strict sequential ordering, so there is no need to explore new transcripts at test time. Although 50Salad transcripts are more diverse, the dataset is very small (40 training videos). Hence, it is challenging for models to learn actions well, and our unconstrained method becomes more sensitive to the exploration rate τ. Having said that, our results on split 1 of 50salad in Table 8 show that with a low exploration rate our method can still outperform SOTA in online segmentation. We chose split1 as our test set because it has the largest sequential variation to the other 4 splits of the 50Salad dataset.

6.4. Qualitative Results

The qualitative comparison in Fig.5 illustrates how our approach for action segmentation and error detection outperforms other constrained baselines. The top video shows an incorrectly assembled record player, which includes two background segments (idle time) and an unfastened screw. During the idle segments, the cosine similarity drops below the threshold, indicating a sequential anomaly. Hence, our proposed Viterbi algorithm is capable of deviating from the seen transcripts and infers these unseen idle segments. Although the Greedy approach detects the idle segments, it produces false positives and over-segmentation. Furthermore, DPcon, conditioned by the training transcripts, not only fails to detect the idle segments, but also mistakenly

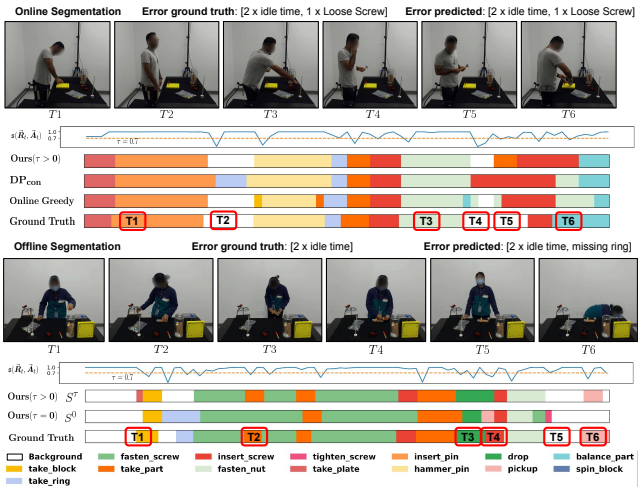

Figure 5. Online and offline segmentation of an incorrectly assembled record player (top) and airplane (bottom). Position of each image in the video is indicated by its corresponding red box.

predicts fastening after the insertion of a screw.

In Fig.5 (bottom), S^0 and S^τ denote constrained and unconstrained offline segmentation of an airplane assembly, respectively. The video contains two instances of idle segments and a situation where a dropped screw is not retrieved until the end of the video. Our segmentation approach captures the sequential anomaly caused by the delay in retrieving the dropped item. In contrast, the action sequence in S^0 only follows one of the training transcripts, considering the second idle time as the end of the video and missing the final pickup. It's worth noting that strictly adhering to the training transcripts can lead to false predictions, such as missing the action "take ring" in the beginning of the video.

7. Conclusion

We presented the ATA dataset as the first anomalous instructional video dataset, and annotated it with temporal and spatial labels. Additionally, we proposed an unconstrained Viterbi algorithm that allows real-time segmentation of anomalous videos into unseen action sequences. We used the segmentation results to introduce a baseline to detect pre-defined human errors, and benchmark results on the ATA dataset. Finally, in our experiments, we showed how our proposed method outperforms SoTA in action segmentation of anomalous videos in two public datasets.

8. Acknowledgment

We would like to send our special thanks to Huan Nguyen, David Fox and Nirav Savaliya for their contributions to this paper and the ATA dataset collection.

References

[1] Andra Acsintoae, Andrei Florescu, Mariana-Iuliana Georgescu, Tudor Mare, Paul Sumedrea, Radu Tudor Ionescu, Fahad Shahbaz Khan, and Mubarak Shah. Ubnormal: New benchmark for supervised open-set video anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20143–20153, 2022. 2

[2] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4575–4583, 2016. 2

[3] Marcella Astrid, Muhammad Zaigham Zaheer, Jae-Yeong Lee, and Seung-Ik Lee. Learning not to reconstruct anomalies. *arXiv preprint arXiv:2110.09742*, 2021. 2

[4] Nadine Behrmann, S Alireza Golestaneh, Zico Kolter, Jürgen Gall, and Mehdi Noroozi. Unified fully and timestamp supervised temporal action segmentation via sequence to sequence translation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pages 52–68. Springer, 2022. 6

[5] Yizhak Ben-Shabat, Xin Yu, Fatemeh Saleh, Dylan Campbell, Cristian Rodriguez-Opazo, Hongdong Li, and Stephen Gould. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 847–859, 2021. 2, 3

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 6

[7] Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. D3tw: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3546–3555, 2019. 2

[8] Xiaobin Chang, Frederick Tung, and Greg Mori. Learning discriminative prototypes with dynamic time warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8395–8404, 2021. 2

[9] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, pages 1–23, 2022. 3

[10] Hanqiu Deng, Zhaoxiang Zhang, Shihao Zou, and Xingyu Li. Bi-directional frame interpolation for unsupervised video anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2634–2643, 2023. 2

[11] Ehsan Elhamifar and Zwe Naing. Unsupervised procedure learning via joint dynamic summarization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6341–6350, 2019. 2

[12] Alireza Fathi, Xiaofeng Ren, and James M Rehg. Learning to recognize objects in egocentric activities. In *CVPR 2011*, pages 3281–3288. IEEE, 2011. 3

[13] Mingfei Gao, Yingbo Zhou, Ran Xu, Richard Socher, and Caiming Xiong. Woad: Weakly supervised online action detection in untrimmed videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1915–1923, 2021. 2, 6, 7

[14] Reza Ghoddoosian, Isht Dwivedi, Nakul Agarwal, Chiho Choi, and Behzad Dariush. Weakly-supervised online action segmentation in multi-view instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13780–13790, 2022. 2, 4, 5, 6, 7

[15] Reza Ghoddoosian, Saif Sayed, and Vassilis Athitsos. Hierarchical modeling for task recognition and action segmentation in weakly-labeled instructional videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1922–1932, January 2022. 2

[16] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 780–787, 2014. 2, 3

[17] Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding*, 163:78–89, 2017. 2

[18] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12066–12074, 2019. 5

[19] Dongha Lee, Sehun Yu, Hyunjun Ju, and Hwanjo Yu. Weakly supervised temporal anomaly segmentation with dynamic time warping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7355–7364, 2021. 2

[20] Sangmin Lee, Hak Gu Kim, and Yong Man Ro. Bman: Bidirectional multi-scale aggregation networks for abnormal event detection. *IEEE Transactions on Image Processing*, 29:2395–2408, 2019. 2

[21] Jun Li, Peng Lei, and Sinisa Todorovic. Weakly supervised energy-based learning for action segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6243–6251, 2019. 2, 4, 5, 6, 7

[22] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European conference on computer vision (ECCV)*, pages 619–635, 2018. 3

[23] Zijia Lu and Ehsan Elhamifar. Weakly-supervised action segmentation and alignment via transcript-aware union-of-subspaces learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8085–8095, 2021. 2, 6, 7

[24] Didik Purwanto, Yie-Tarng Chen, and Wen-Hsien Fang. Dance with self-attention: A new look of conditional random fields on anomaly detection in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 173–183, 2021. 2

[25] Yicheng Qian, Weixin Luo, Dongze Lian, Xu Tang, Peilin Zhao, and Shenghua Gao. Svip: Sequence verification for procedures in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19890–19902, 2022. 2, 3, 6

[26] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 754–763, 2017. 2

[27] Alexander Richard, Hilde Kuehne, Ahsan Iqbal, and Juergen Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7386–7395, 2018. 2, 5

[28] Nicolae-Cătălin Ristea, Neelu Madan, Radu Tudor Ionescu, Kamal Nasrollahi, Fahad Shahbaz Khan, Thomas B Moeslund, and Mubarak Shah. Self-supervised predictive convolutional attentive block for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13576–13586, 2022. 2

[29] Marcus Rohrbach, Anna Rohrbach, Michaela Regneri, Sikandar Amin, Mykhaylo Andriluka, Manfred Pinkal, and Bernt Schiele. Recognizing fine-grained and composite activities using hand-centric features and script data. *International Journal of Computer Vision*, 119(3):346–373, 2016. 2, 3

[30] Saif Sayed, Reza Ghoddoosian, Bhaskar Trivedi, and Vassilis Athitsos. A new dataset and approach for timestamp supervised action segmentation using human object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3132–3141, June 2023. 3

[31] Fadime Sener, Dibyadip Chatterjee, Daniel Shelepov, Kun He, Dipika Singhania, Robert Wang, and Angela Yao. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21096–21106, 2022. 2, 3

[32] Yaser Souri, Yazan Abu Farha, Fabien Despinoy, Gianpiero Francesca, and Juergen Gall. Fifa: Fast inference approximation for action segmentation. In *DAGM German Conference on Pattern Recognition*, pages 282–296. Springer, 2021. 2

[33] Yaser Souri, Mohsen Fayyaz, Luca Minciullo, Gianpiero Francesca, and Juergen Gall. Fast weakly supervised action segmentation using mutual consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2, 6, 7

[34] Sebastian Stein and Stephen J McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 729–738, 2013. 2, 3, 8

[35] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1207–1216, 2019. 2

[36] Kamalakar Vijay Thakare, Yash Raghuwanshi, Debi Prosad Dogra, Heeseung Choi, and Ig-Jae Kim. Dyannet: A scene dynamicity guided self-trained video anomaly detection network. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5541–5550, 2023. 2

[37] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269, 1967. 2

[38] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Zhengrong Zuo, Changxin Gao, and Nong Sang. Oadtr: Online action detection with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7565–7575, 2021. 6, 8

[39] Jhih-Ciang Wu, He-Yen Hsieh, Ding-Jie Chen, Chiou-Shann Fuh, and Tyng-Luh Liu. Self-supervised sparse representation for video anomaly detection. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIII*, pages 729–745. Springer, 2022. 2

[40] Guang Yu, Siqi Wang, Zhiping Cai, Xinwang Liu, Chuanfu Xu, and Chengkun Wu. Deep anomaly discovery from unlabeled videos via normality advantage and self-paced refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13987–13998, 2022. 2

[41] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint pattern recognition symposium*, pages 214–223. Springer, 2007. 6

[42] Muhammad Zaigham Zaheer, Arif Mahmood, Marcella Astrid, and Seung-Ik Lee. Claws: Clustering assisted weakly supervised learning with normalcy suppression for anomalous event detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 358–376. Springer, 2020. 2

[43] M Zaigham Zaheer, Arif Mahmood, M Haris Khan, Mattia Segu, Fisher Yu, and Seung-Ik Lee. Generative cooperative learning for unsupervised video anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14744–14754, 2022. 2

[44] Yuansheng Zhu, Wentao Bao, and Qi Yu. Towards open set video anomaly detection. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV*, pages 395–412. Springer, 2022. 2

[45] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3537–3545, 2019. 2